TIES442 - Tasks - Week 3

Deadlines for handing in the tasks, and their effects on the awarded points:

- max 3 points: 7 April 2015 10:15 am
- max 2 points: 14 April 2015 10:15 am
- max 1 point: 30 June 2015 12:00 pm

Grading principles, maximum points for solution:

- completely empty or missing solution (by 30 June): 0p
- 2, 3 or more tasks with no answer at all: 1-2p
- 2 or 3 tasks (from 2, 3b, 3c) that are missing the descriptions of algorithm phases: 2p
- well reasoned solutions to all tasks, including descriptions of all algorithm phases that are more or less correct (2, 3b, 3c): 4p

---

The tasks will appear latest 24 March 10 am. Otherwise, the deadlines will be moved forward as well.

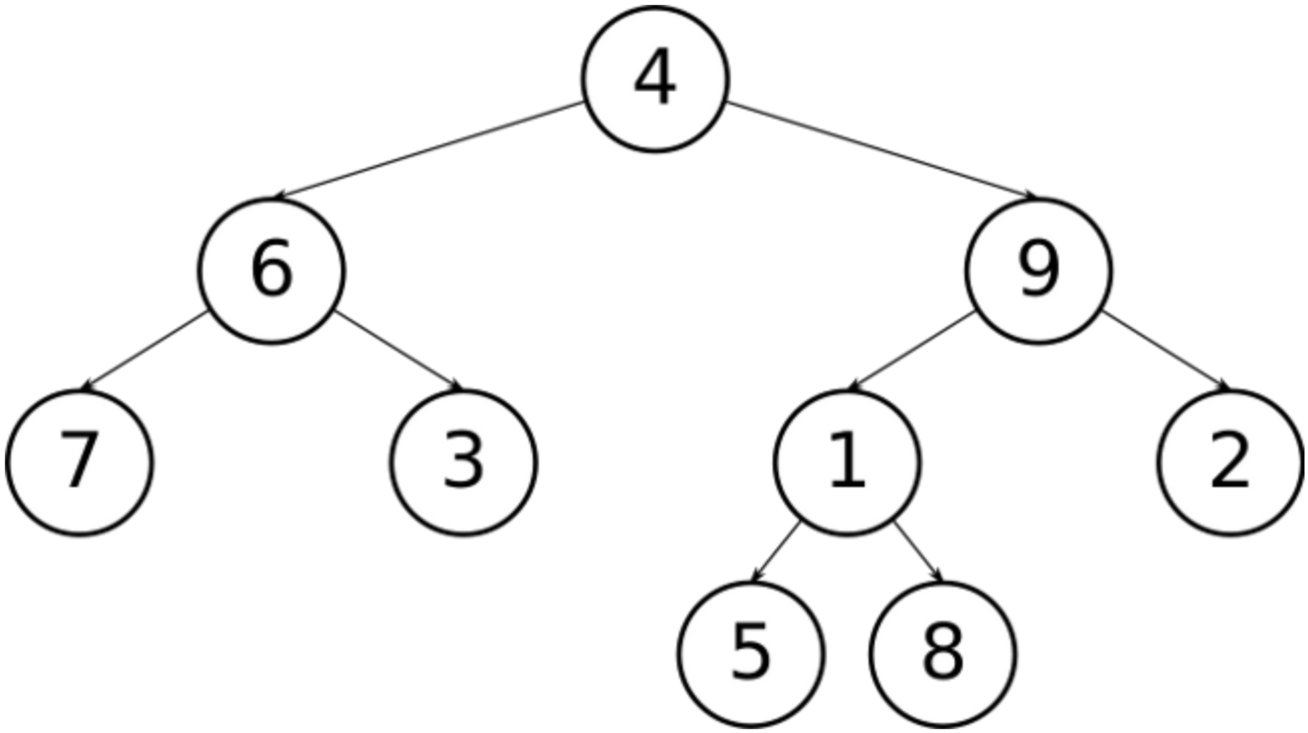
Week 3 is exceptional, because we won't have a lecture on Monday due to Easter. The topic is review of search methods in basic data structures and refreshing memory of concepts related to complexity of algorithms. These are basic terms, and lots of information can be found from the web.

Task 1

Give some feedback on the course so far. What is good, what is bad, would you like to change or improve something? Are there inaccuracies or deficiencies in the materials? Would you like to have some specific topic discussed on the course? You can give your feedback either anonymously on the course page http://ties442.it.jyu.fi (in which case it's enough to write here that this has been done) or in your own name by writing it in the solution document.

Task 2

Let us consider finding the node with label 5 from the following **tree**.



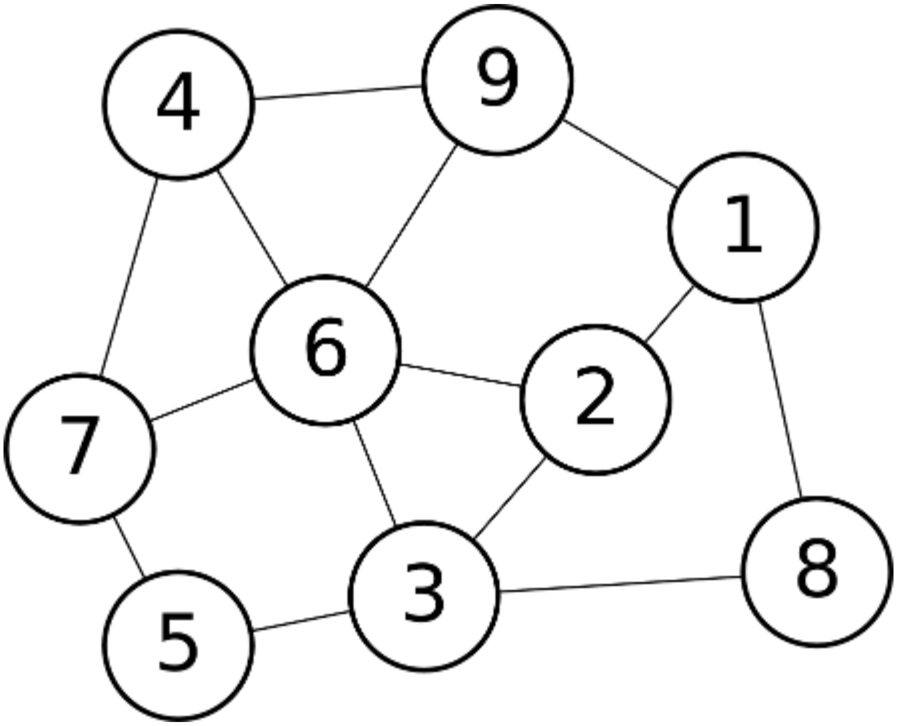
Tree

a. How would **breadth-first search** proceed, meaning in which order the nodes would be examined? What auxiliary data structure could be used in implementing the algorithm? When is this kind of search problematic?

b. How would **depth-first search** proceed, meaning in which order the nodes would be examined? What auxiliary data structure could be used in implementing the algorithm? When is this kind of search problematic, and could these problems be avoided somehow?

Task 3

Let us consider the following **graph**. It consists of **nodes** (or vertices) that are connected to each other by **edges** (or arcs). Find basic information on graph theory using keywords such as '*introduction to graph theory*' or '*graph theory primer*'.



Graph

a. Explain what means the **degree** of a node in a graph, and what does the **diameter** of a graph mean. What is the degree of node 5, and what is the diameter of the graph? What are the neighbors of node 5?

b. How could you search for node 5 in the graph if you do not know its structure, meaning that you get a random starting node (such as 1) and can only ask for the neighbors of a given node? List the phases of the algorithm, meaning the node that is examined at each phase and its neighbors; also describe what was the basis for choosing the next node. What kinds of problems searching in graph can cause, and how could you avoid them?

c. Figure out what is a **spanning tree** of a graph, and what is the **minimal spanning tree**. Give an example of a spanning tree of the above graph. How could you go about finding the minimal spanning tree? (Hint: how could we assign weights to the graph edges?) Describe the phases of the algorithm that finds the minimal spanning tree, meaning which nodes and edges are processes at each phase and what are the grounds for making the choices.

d. How could you use a spanning tree in search operations?

Task 4

Let us consider on high level finding the shortest path between two nodes of a graph, such as nodes 1 and 7 in the above graph. Would it be possible to present this as a search in some simpler data structure by generating it on the fly as search progresses?

Task 5

Refresh your memory and explain what the **O-notation** (Big-Oh), for example $O(n)$ or $O(n^2)$ when speaking of complexity of algorithms. What does it mean that a problem is **NP-hard**? What does it mean if it is **NP-complete**?
